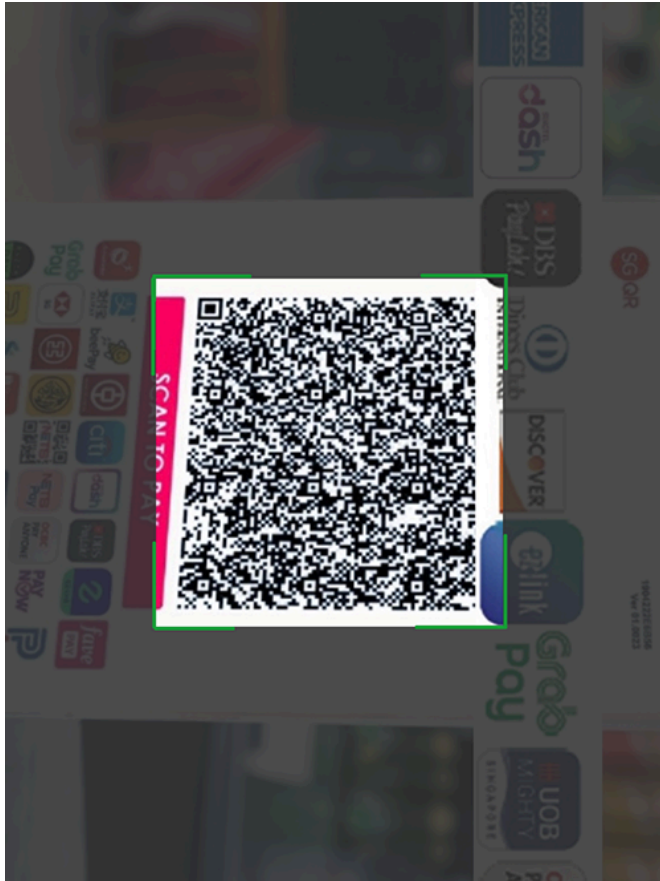


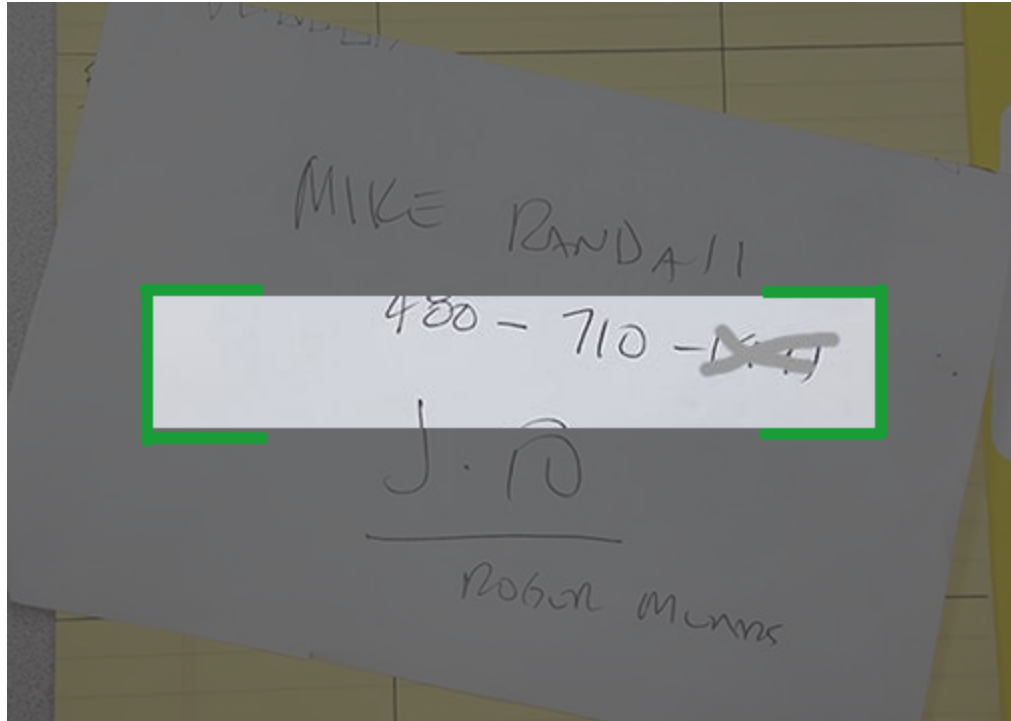
### **ТЗ на создание программного модуля «Виджет сканирования QR кодов и номеров телефонов»**

Необходимо реализовать функционал сканирования и распознавание значений объектов, отображаемых в зоне видимости камеры в реальном времени в среде всех последних версий браузеров во всех операционных системах: Windows 10, MacOS, Adndroid, iOS. Общий принцип работы ПО следующий:

1. По нажатию на иконку старта сканирования запускается камера устройства. Если браузер поддерживает управление подсветкой устройства, и картинка слишком темная, нужно включить фонарь/подсветку и иметь иконку на экране переключающую подсветку.
2. Всего для первой версии приложения нужна поддержка двух режимов:
  - a. Распознавание QR кодов
  - b. Распознавание номеров телефона (как печатных, так и рукописных)
3. Сканеры включаются независимо друг от друга, нажатием разных иконок в нижней панели экрана – иконка с QR кодом и иконка с мобильным телефоном. В режиме реального времени нужно производить поиск соответствующего объекта в зоне видимости камеры, выделенной видеоискателя (уголками прямоугольника) с использованием соответствующих инструментов:
  - a. Для QR кода: в центре зоны видимости камеры видимость прозрачная, за пределами затемненная, есть уголки, обозначающие зону Рамки видеоискателя, куда пользователь должен поместить QR код

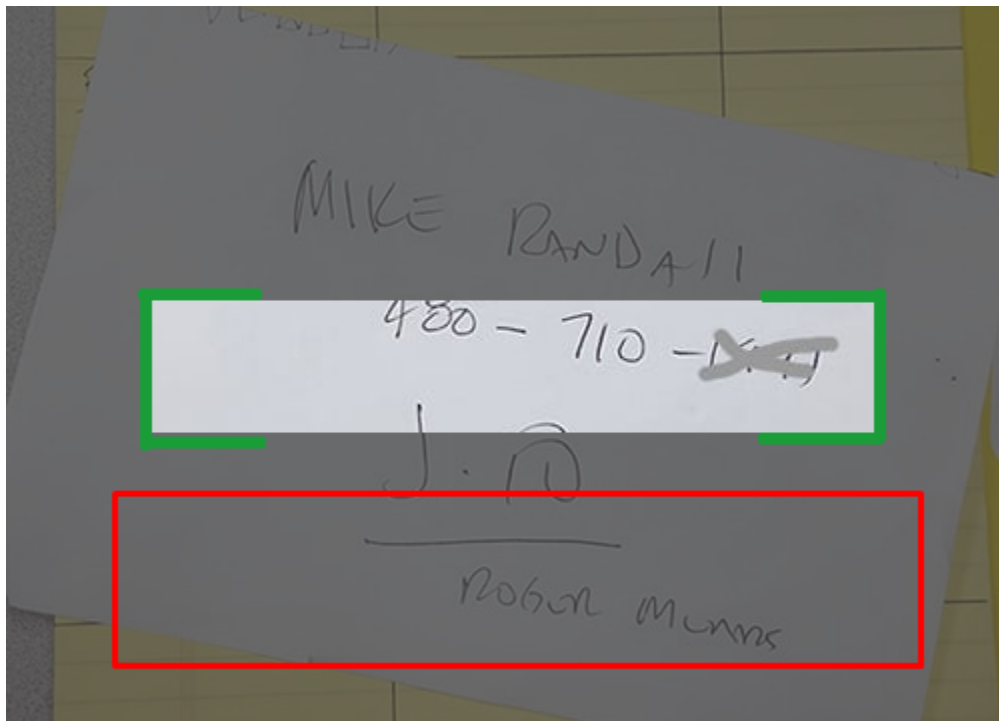
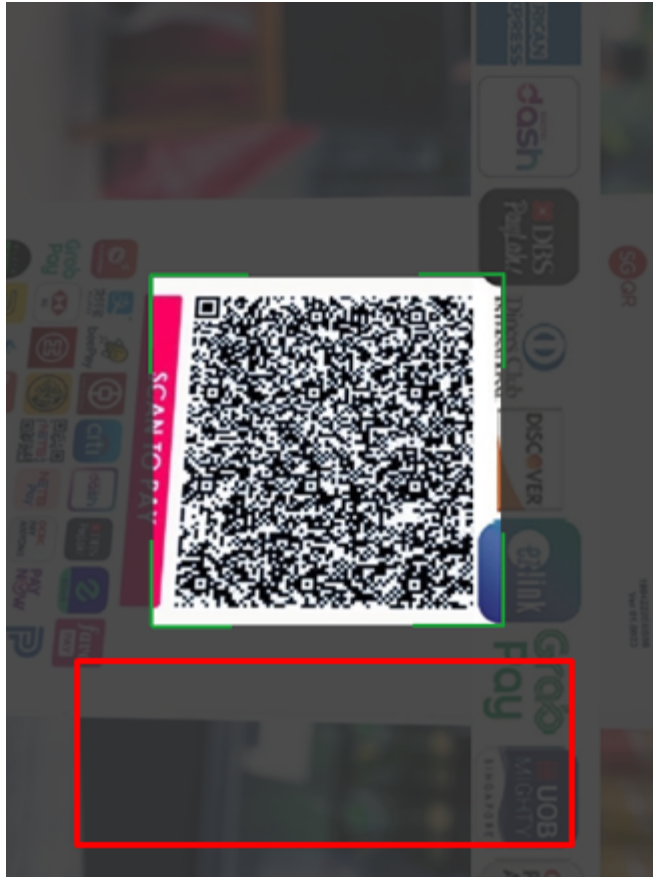


- b. Для номера телефона аналогично, но с другими пропорциями рамки видеоискателя:



Пропорции и элементы указаны приблизительно и являются предметом настройки для получения наибольшей производительности и удобства для пользователя. Калибровка будет производиться с учетом технологических требований, определяемых заказчиком.

4. В момент сканирования должен производиться инференс полученных с камеры изображений с использованием нейронной сети (фреймворк Tensor.js, все работает в самом браузере). На инференс, должно подаваться только изображение, вырезанное из зоны видимости Рамки видеоискателя, описанной выше. Частота инференса должна учитывать производительность устройства, то есть если девайс успевает обрабатывать 3 кадра в секунду – остальные кадры из видеопотока камеры должны пропускаться.
5. При определении положительного ответа от нейронной сети (обнаружении необходимого объекта, возвращение координат символов и т.п.) запускается последующая цепочка обработки данных другими нейронными сетями для получения конечного текстового результата. Результат необходимо выводить в выделенных зонах текстом:



6. Помимо вывода распознанного значения на экран должен вызываться callback метод с отправкой этих значений вызывающему виджет коду.
7. Для устройств, поддерживающих GPU ускорители на борту мобильного устройства необходимо производить инференс с максимальной производительностью – то есть с их обязательным использованием
8. Нужно иметь режим DEBUG с выводом на экран показателей производительности всего процесса:
  - a. FPS отрисовки изображений с камеры на канвас
  - b. FPS инференса нейронной сети
  - c. Возможны дополнительные параметры по согласованию
9. После получения отлаженного решения необходимо сделать компонент на React Native, который может подключаться к любым нативным приложениям на iOS Android
10. Весь код должен быть полностью по стандартам best practices
11. В результате заказчик ожидает получить:
  - a. Репозиторий в GitHub и исходным кодом на React.js виджета сканирования и распознавания QR кодов и номеров телефонов с подробной, оформленной для передачи сторонним клиентам инструкцией по его интеграции в React Native приложение
  - b. Приложение для iOS в приложении TestFlight с демонстрацией подключенного модуля и получением в само приложение значений через callback
  - c. Приложение для Android в виде APK файла с демонстрацией подключенного модуля и получением в само приложение значений через callback

Рекомендуемые для рассмотрения библиотеки и фреймворки:

Для обработки QR кода будет полезной статья с примерами и готовым кодом <https://medium.com/js-now/creating-a-real-time-qr-code-scanner-with-vanilla-javascript-part-1-2-creating-the-scanner-a8934ee8f614>. Также необходимо провести ресерч и поиск последних разработок в этой области.

Для распознавания телефонов заказчик предоставит нейронную сеть, находящую в зоне видимости символы и возвращающую координаты этих символов на изображении. Далее произвести распознавание символов необходимо с использованием фреймворка, описанного по ссылке <https://codelabs.developers.google.com/codelabs/tfjs-training-classfication>. Также необходимо провести ресерч и поиск последних разработок в этой области.